The Impact of Hyperledger Fabric Setup on Blockchain Performance when Using Large Volumes of Heterogeneous Medical Data*

Ana Caroline Fernandes Spengler[1], Paulo Sérgio Lopes de Souza[1]

[1] Instituto de Ciências Matemáticas e de Computação - Universidade de São Paulo (USP)
São Carlos, SP – Brazil

anacspengler.acs@gmail.com, pssouza@icmc.usp.br

Abstract. *Blockchain can be seen as a data distribution tool that guarantees immutability. As its use continues to expand across various sectors, it becomes increasingly important to investigate Blockchain's performance concerning its different components and data originating from diverse application domains. This study explores the blockchain ecosystem, focusing on block creation, validation, network size, and partition processes. The chosen methodology involves utilizing Hyperledger Fabric for sharing medical information. To assess performance, Hyperledger Caliper was employed to collect throughput and latency. Among the key findings, we show that segregating the network into channels impacts the performance of Blockchain, mainly when the number of participating nodes increases. Sizes and timeouts to create new blocks influence the system's performance. This paper contributes to developers by highlighting factors impacting blockchain-based applications' performance.*

1. Introduction

With the growing prominence of Bitcoin and other cryptocurrencies, the use of blockchain as a tool for decentralized data distribution is consolidating, especially for financial applications [Fang et al. 2022]. Moreover, blockchain technology is being applied worldwide in other sectors such as commodity distribution, goods transactions, education, and healthcare. The healthcare sector, in particular, has been a blockchain's target by allowing distributed access to medical data, addressing the need for data interoperability. Managing distributed medical data has significant challenges. It involves sharing heterogeneous data with privacy and security requirements, owing to the sensitive nature of the data. Blockchain offers such features to applications, and additionally, these applications enable institutions and patients to jointly take responsibility for the data, negating the need for third-party control over data access and distribution.

This ongoing research project evaluates the Fabric blockchain's performance when applied to heterogeneous medical data. Our early results show the Fabric's performance under distinct insertion and read demands of data [Spengler and Souza 2021a], and also the CouchDB database's [Foundation 2020] impact on data storage [Spengler and Souza 2021b]. These earlier studies revealed that employing a database within the blockchain is beneficial in scenarios with a low insertion request rate. Developers should carefully analyze the blockchain structure if a higher demand exists to

*The authors thank the financial support of the FAPESP through the process 2019/26702-8.

mitigate potential performance losses. This paper presents whether other factors within a blockchain network can impact its performance, even with database use.

We execute experimental studies utilizing Hyperledger Fabric [Hyperledger 2019], Hyperledger Caliper benchmark [Project 2020] for network throughput and latency data, and the CouchDB nonSQL database [Foundation 2020]. We analyze block creation, network consensus, network size, and node distribution. The heterogeneous data used in the experiments are based on the medical database MIMIC-III, which compiles data from a Boston-based hospital [Johnson et al. 2016].

Our main result reveals that segregating the network into channels impacts the performance of Blockchain, mainly when the number of participating nodes increases. This paper can contribute to blockchain application developers by highlighting critical aspects that impact the performance of such applications.

This paper follows a seven-section structure. Section 2 presents our related works. Section 3 delves into blockchain particulars. The experimental design is outlined in Section 4. Sections 5 and 6 elaborate on each experiment's specifics, the resulting outcomes, and the subsequent analysis. Finally, Section 7 offers our concluding remarks.

2. Related Work

Al-Sumaidaee et al. (2023) propose a blockchain test network built on Hyperledger Fabric to optimize the flow of information between two medical institutions where one does not trust the other [Al-Sumaidaee et al. 2023]. They utilize a decentralized network to enhance the management of shared information.

Pajooh et al. (2022) introduce a scalable framework for real-time monitoring of large-scale IoT systems, supported by Hyperledger Fabric [Honar Pajooh et al. 2022]. This framework is employed in an experimental performance analysis of a network, using the metrics throughput, latency, network size, scalability, and the number of peers served by the distributed platform.

Chung et al. (2019) optimize blockchain for scalability in industry applications based on Hyperledger Fabric with CouchDB for heterogeneous data [Chung et al. 2019]. Unlike our work, Chung et al. (2019) did not use a real-world database.

Roehrs et al. (2019) evaluate data distribution using metrics like response time, memory usage, and CPU utilization in a custom blockchain implementation. The study lacks the support of development platforms like Hyperledger Fabric [Roehrs et al. 2019]. Their experiments simulated concurrent access from 40,000 clients, and the authors conclude that blockchain results in low response times and increased data availability.

Shen et al. (2019) analyze the security and efficiency of distributed storage of medical data using blockchain and cloud storage tools like MedChain (which includes a blockchain module, a directory module, and a client) [Shen et al. 2019]. The paper showcases overhead in different data-sharing scenarios. Unlike our work, Shen et al. (2019) primarily evaluate MedChain in terms of information security and application efficiency.

Baliga et al. (2018) characterize the performance of Hyperledger Fabric using the metrics throughput and latency, employing the benchmark Hyperledger Caliper [Baliga et al. 2018]. Aspects of the platform impacting Hyperledger Fabric's performance and distributed applications built on it are considered. Unlike our work, this paper does not compare performance while considering database usage, such as CouchDB.

Dinh et al. (2017) compare platform performance using the BlockBench framework [Dinh et al. 2017]. They evaluated Hyperledger, Ethereum, and Parity platforms using throughput, latency, and fault tolerance metrics. The authors note that their results need to be generalized and that further executions and analyses are needed.

3. Blockchain

Blockchain network allows the development of decentralized systems using Peer-to-Peer (P2P) networks, storage of data in a chain format (blockchain), consensus rules for validating transactions, and a decentralized global authentication system to validate the records (e.g., the Proof-of-Work (PoW) algorithm) [Antonopoulos 2017]. Transactions and records are stored in blocks representing the blockchain's basic structure.

Blocks are linked in so that the content of one block depends on the previous block, forming a chain. This sequence of blocks is spread in a P2P distributed network. The creation of blocks is called mining. PoW is the computational problem employed for the production of bitcoins. This algorithm calls a process of trial and error to find a predetermined result. There is a high computational cost for its execution; however, verifying the outcome requires low computational power. Newly created blocks are propagated in the network and copied locally in the chain of participating nodes. The node verifies each transaction stored by the block; if all transactions have been validated, the node adds the block to its local copy of the blockchain. In this way, the validation process takes place in two steps: in the transactions and the block; once inserted in the chain, the data is immutable. The application determines the validity of a transaction [Antonopoulos 2017]. These features allow blockchain to ensure the distribution of information in a decentralized and secure environment. Even if participants do not trust each other, they can exchange data without a centralizing agent [Antonopoulos 2017].

We chose the platforms Hyperledger Fabric and the Hyperledger Caliper benchmark for collecting performance metrics. Although the Hyperledger Fabric architecture contains the same elements described above, it presents some differences. One of these differences is that the distributed chain comprises two storage components: the world state database and the blockchain. The first characterizes the chain's state at a given time by storing data in a key-value format in LevelDB or CouchDB databases. The blockchain records each transaction that changed the world state as a global value [Hyperledger 2019].

Smart contracts in the Hyperledger Fabric are associated with a transaction endorsing that determines how many and which nodes have to approve a transaction. All transactions are stored in the chain, whether valid or not, but only valid transactions change the state world. Once distributed across the network, transactions are validated in two steps: (1) it verifies that the transaction has followed the corresponding endorsing policy, i.e., that the organizations have validated it, and (2) it verifies whether the value in the state world matches the transaction being evaluated [Hyperledger 2019].

A network created within Hyperledger Fabric includes an Ordering Service to establish the order of transactions and group them into blocks. The Orderer node implements the Ordering Service. Keeping the chain stored on each node consistent is essential, allowing nodes to focus on validating and storing transactions on the chain without ordering them [Hyperledger 2019]. Hyperledger Fabric has two transactions: query and update. Query is simpler, consisting of the search for a value in the blockchain, not requiring the validation and consensus process of the network. The updated transaction changes the chain, requiring the validation of the rest of the network participants [Hyperledger 2019].

Caliper allows obtaining results on the number of transactions accepted in the chain , throughput, and latency. It generates a workload for evaluating the system and continuously monitors its response, generating a report. It uses configuration files, defining the benchmark characterization, network configuration, and workload [Project 2020].

4. Experimental studies

The two experiments described in this paper evaluate the performance of a blockchain network as a safe infrastructure for the distribution and storage of heterogeneous data. The first focuses on the consensus and mining processes, and the second on the network configuration. Their object of study is the Hyperledger Fabric (v. 1.4.1). The quality focus is to verify Fabric's performance with heterogeneous medical data. The perspective is to quantify Fabric's performance in aspects already known as blockchain bottleneck, helping developers of Hyperledger Fabric-based applications.

4.1. Planning

The two experiments used medical data from the MIMIC-III database collected from clinical patients admitted to the Beth Israel Deaconess Medical Center in Boston, Massachusetts. The database is divided into 26 files containing information regarding hospital patients, ICU stays, hospital records, and dictionary files [Johnson et al. 2016]. Four files were chosen, one from each category above, using as selection choice criteria the number of columns of each file and the necessity of an index.

The PATIENTS file keeps the registers of the hospital's patients. It has eight columns and does not require the creation of an index to insert data. The size of each entry can be up to 50 bytes. The D_ITEMS file contains ten columns and demands two index creations for insertion. Each entry can be up to 608 bytes. The PRESCRIPTIONS file has 19 columns, and its insertion process requires five index creations. The entries can be up to 1592 bytes. The INPUTEVENTS_MV file includes 31 columns and takes four index creations. The columns for this file can add up to 826 bytes.

The metrics chosen to evaluate the performance of the Fabric blockchain were throughput and latency, which were calculated using the benchmark Hyperledger Caliper. The benchmark considers two types of transactions: read and update. The first one does not change the chain and allows access to the data in the chain. The second modifies the network's chain, using the validation process for insertion.

For read transactions, latency is the response time from when a transaction is received minus when it is submitted. The throughput of this type of transaction is calculated as the total number of transactions processed by the whole time in seconds. For update transactions, latency is the time from when transactions are submitted until their results are known to the network, incorporating the transaction validation and propagation times according to the consensus policy. The update transaction throughput is the rate at which valid transactions are submitted to the blockchain by the total time in seconds it takes to validate the transaction across the entire network.

4.2. Execution

The Hyperledger Fabric version 1.4.1 used in this study does not feature an algorithm like Bitcoin's Proof-of-Work for consensus protocol. The ordering service is responsible for ordering the transactions and generating the block. The Solo implementation of the ordering service was used in our experiments.

The Caliper benchmark generated the workload for the five operations performed (four insertions and one search). Data is read from the MIMIC-III files, distributed over the network, and stored in CouchDB for each insertion. The nodes validate the insert transactions during the transaction distribution process (see Section 3). The data insertion operations are the first to be executed; the first insertion is from the PATIENTS' file, followed by D_ITEMS, PRESCRIPTIONS, and INPUTEVENTS_MV. The last operation executed is the search, which relies on inserting the PATIENTS file. The different operations are executed in blocks, where a task only starts after finishing the previous one.

The platform utilized in our experiments has this configuration: Operating System Ubuntu versions 16.04.7 LTS e 20.04 LTS, CPU QEMU Virtual CPU (512 KB Cache, 4 GHz), Memory 16 GB RAM and Hard Disk Seagate HD 2TB ST2000DM001-1ER1.

5. Experiment 1 - Transaction validation and block building

This experiment evaluates how consensus protocol and the block-building process can affect transaction latency and throughput in Hyperledger Fabric. This experiment verifies three hypotheses: (1) Decreasing the number of nodes required to validate a transaction improves the performance of data insertions in terms of latency and throughput; (2) Creating larger blocks decreases network performance concerning latency and improves the performance of the system's throughput; and (3) Increasing the time to create a block decreases the system performance for latency and improves performance for throughput.

Four factors were analyzed in this experiment. Two refer to the consensus process: the endorsing policy (levels 'AND' and 'OR') and the number of endorsing peers (levels 2 and 4). The first hypothesis investigates both. The other two factors are involved in the building block processes, block size (levels 10 and 500) and block timeout, representing the maximum time to generate a block (levels 1s and 10s.) The ordering service determines rules for the creation of blocks. A block is created when it reaches its size or a timeout happens. The second and third hypotheses inspect the last two factors.

The consensus process is based on the validation policy and the number of endorsing nodes in the network. The OR policy defines that the transaction must be validated by at least one node of either organization, thus consuming the resource of only one node of the system in total. The AND policy states that transactions must be validated by at least one node from each organization in the network. The number of endorsing nodes quantifies how many nodes are required to validate a transaction. This value refers to the first step of the validation process for update-type transactions.

When the validation policy is AND and 4 nodes are selected, two nodes in each organization must guarantee consensus. For the OR policy, only 1 of the nodes must indicate to the requesting client if the transaction is valid or not. When considering 4 endorsing nodes, all network peers can validate a transaction. Only those selected nodes can return on the transaction's validity when set up with two validating nodes.

The network of this experiment consists of 9 Docker containers. Five nodes in the blockchain network are running Hyperledger Fabric, and 1 is selected as the orderer node. The other 4 are participants in the network split into 2 organizations with 2 peers each. Four docker containers are hosting CouchDB.

5.1. Results

Figures 1 to 5 show the values found for the latency and throughput of the executions. In the average latency graphs, the error bar considers the standard deviation of the operations.

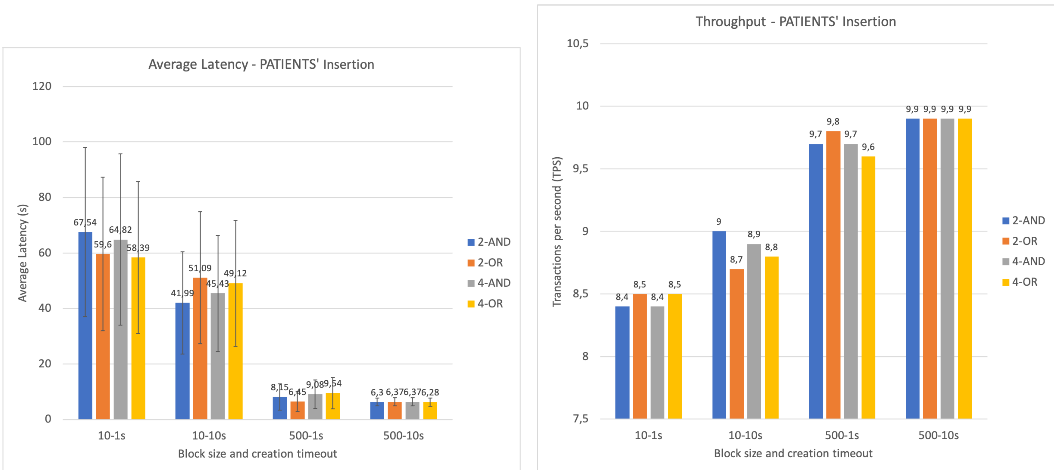

Figure 1. PATIENTS

The endorsing policy and the number of endorsing peers have negligible influence on the application's performance. There were scenarios in which the AND policy resulted in lower latency and higher throughput than the OR policy, contrary to what was anticipated. This oscillation concerning the expected behavior occurs in a few scenarios, and the values are within the standard deviation observed, as shown in Figure 1 with 4 endorsing nodes, block size 10, and 10 s timeout.

In contrast, the results found in the evaluated scenarios show the impact of changing block size and timeout. Figure 2 shows how the application performance improves with each change of these parameters to decrease latency and increase throughput. This performance improvement is most evident when inserting files with more fields for indexing into the CouchDB database, as in Figures 3 and 4. The average latency and throughput values remained unchanged in all scenarios tested. Changing the consensus process and block creation was expected to keep the metrics evaluated for the search since this type of transaction does not require such steps to occur.

5.2. Analysis

The first hypothesis says that reducing the number of peers involved in the endorsing processes increases the throughput and decreases the system's latency. Our results in this experiment show that this first hypothesis is invalid, as we needed to verify a consistent

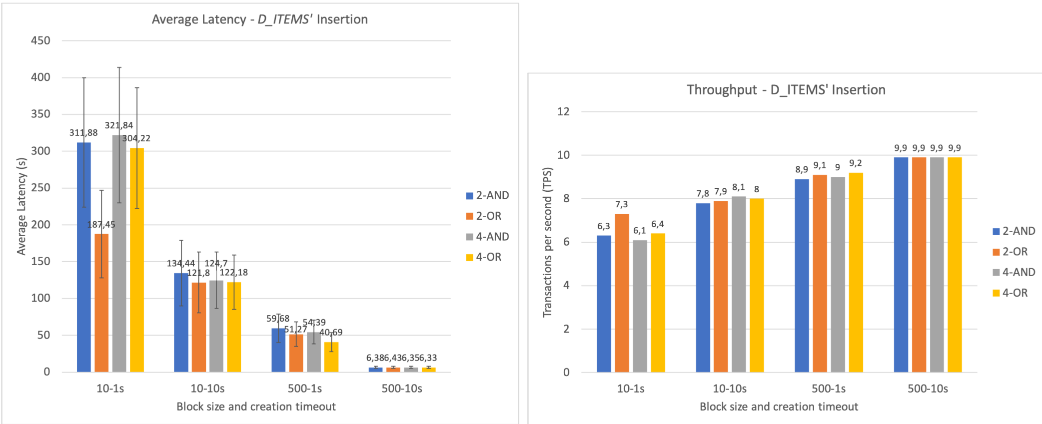

Figure 2. D_ITEMS

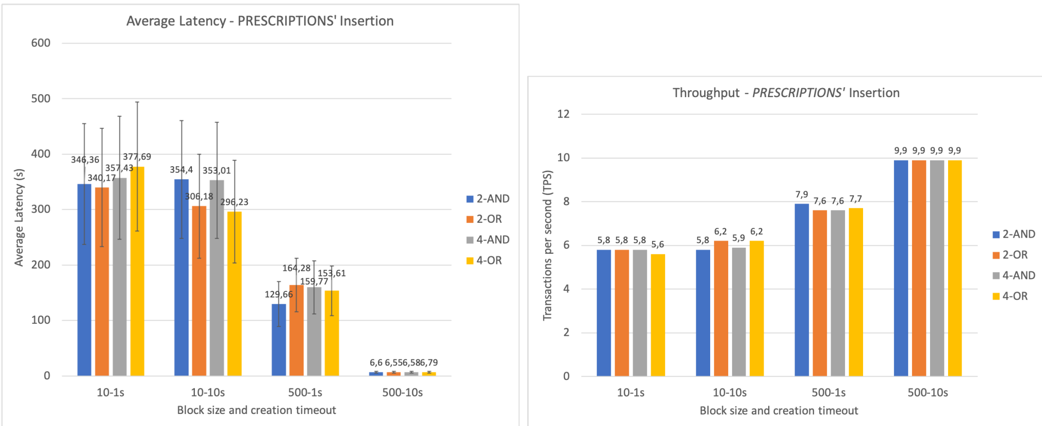

Figure 3. PRESCRIPTIONS

behavior of increased throughput and reduced latency in these cases. The results indicate that using CouchDB must have impacted the validation process and that this behavior needs to be better investigated in future research.

The second research hypothesis affirms that creating larger blocks increases latency (worsting the performance) and increases the throughput of insertion operations (making the performance better). The expectation was that creating wider blocks would take more time. However, it would concurrently spread a more significant number of transactions through the blockchain's network once the block is ready. Contrary to the hypothesis, the results demonstrate that the average latency decreases when the network is executed with a larger block size, i.e., the latency is longer when small blocks are considered. These results suggest a network overhead when a more considerable number of small blocks are created compared to fewer larger ones. As expected in our hypotheses, the throughput increased in those scenarios.

The third hypothesis refers to the block-building time. Expanding this interval increases the system's latency and throughput. Our results illustrate how increasing the time limit for generating a new block decreased latency and improved throughput.

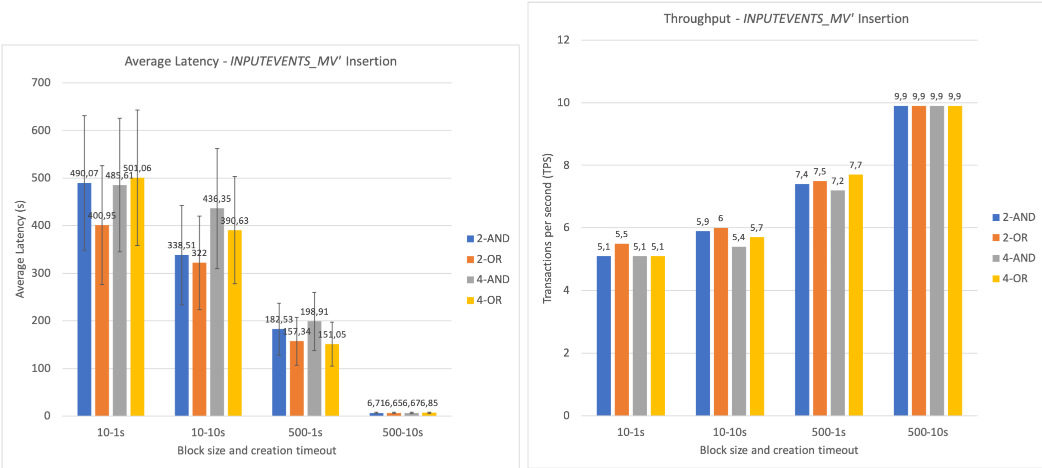

Figure 4. INPUTEVENTS_MV

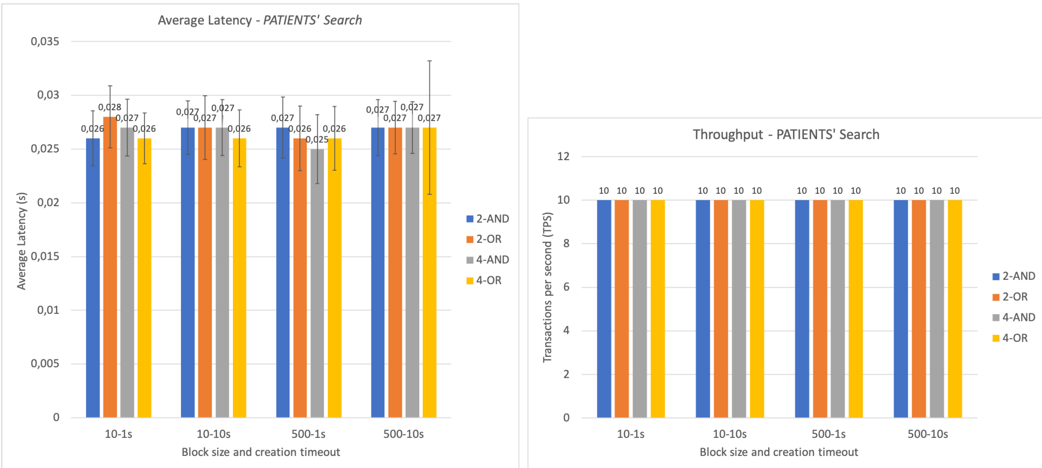

Figure 5. SEARCH

6. Experiment 2 - Hyperledger Fabric Network Configurations

The second experiment analyzes the effect of different network configurations on the performance of the Hyperledger Fabric with CouchDB. The hypotheses of this experiment are: (1) Increasing the number of nodes, and thus the number of machines in the network, deteriorates the performance in terms of the average latency and throughput in blockchain applications; (2) Increasing the number of organizations worsens the performance of the metrics collected; i.e., it increases latency and decreases throughput. The second hypothesis focuses on Hyperledger Fabric's architecture, which limits communication between nodes that do not belong to the same organization. Inter-organization communication is performed only by anchor nodes. An organization needs at least one anchor node.

The two factors are the Number of Peers (levels 4, 8, and 12) and the Number of Organizations (levels 1, 2, and 4.) For each number of nodes tested (4, 8, and 12), three networks were developed, with 1, 2, and 4 organizations. Besides the nodes participating in the network, each network has a node of the orderer type. CouchDB was considered in all runs performed. Hence, all machines had at least two containers, one representing the network node and another with CouchDB associated with the node. The number of clients in the network equals the number of nodes within each organization. In all execution scenarios, the nodes are in a single channel.

6.1. Results

Figures 6 to 9 show our results by modifying the number of nodes in the blockchain and the number of organizations in which these nodes are distributed. The graphs of average latency contain the standard deviation for this metric. The results show that increasing the number of organizations for each number of nodes tested (4, 8, and 12 network nodes) led to higher average latency and lower throughput in every tested scenario. In most runs, the more significant number of nodes also generated higher latency and lower throughput when compared in isolation to networks with 1, 2, and 4 organizations.

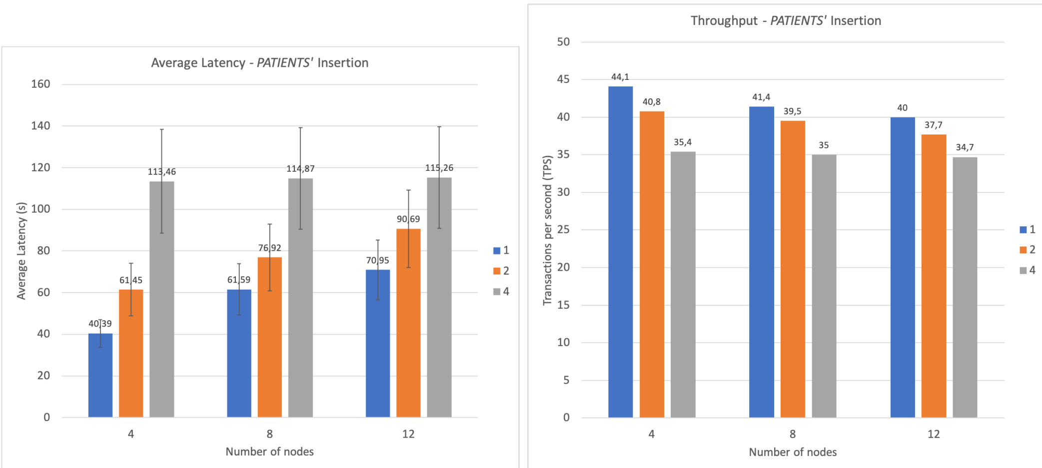

Figure 6. PATIENTS

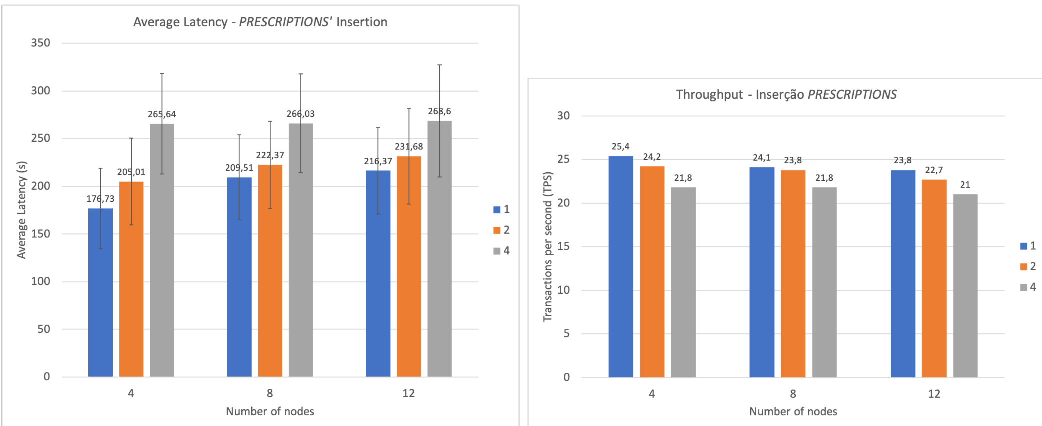

Figure 7. PRESCRIPTIONS

However, there were execution scenarios where the average latency of a network with fewer nodes exceeded the latency of the network with more nodes for the same number of organizations. As in the PRESCRIPTIONS file insertion (Figure 7), the average latency reduced when comparing the network with 8 and 12 nodes, both with 1 organization each. Another latency decrease scenario with the same file was between the network of 4

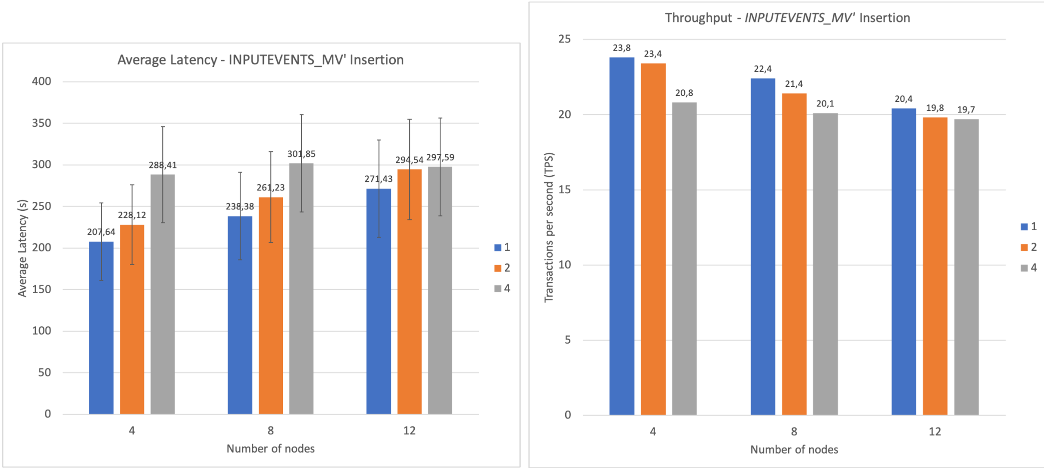

Figure 8. INPUTEVENTS_MV

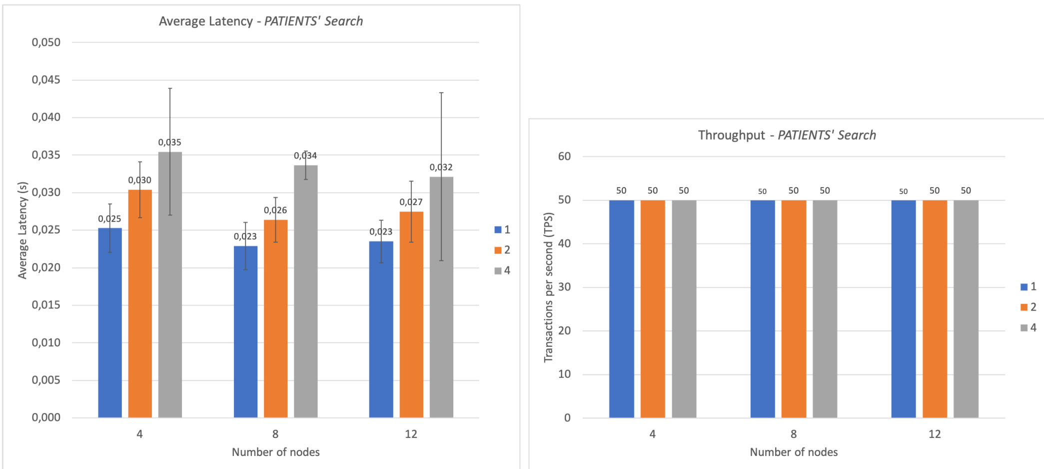

Figure 9. SEARCH

and 8 nodes with 4 organizations. This also occurs in inserting the INPUTEVENTS_MV file (Figure 8) in executions with the 4 and 8-node network for 4 organizations each.

Figure 9 shows the average latency for the search operation, which obtained similar values given the variation in the factors analyzed. The only tendency observed is the increase in average latency with a higher number of organizations for the same number of nodes. The search requests are answered by the nodes belonging to the same organization as the client that made the request. With only one organization in the network, it is possible to answer the demand faster since more nodes are available for this function.

6.2. Analysis

The first hypothesis establishes the relationship between increasing the number of nodes and a decrease in the performance of the blockchain network. The results show a tendency for network performance to deteriorate by increased latency and decreased throughput when the number of nodes rises. However, as pointed out in the section 6.1, there are scenarios in the insertion operations where the network with fewer nodes surpasses the latency of larger ones. In those scenarios, the network throughput was lower or of similar value, as expected. Falling latency can indicate that more organizational nodes positively impact this metric. This outcome may occur because there is no barrier in communicating nodes from the same organization, leading to quicker responses.

The second hypothesis establishes that increasing organizations worsens the performance of the blockchain network by increasing latency and decreasing throughput. Our results showed that the hypothesis is true for latency, but for throughput, the results varied for the two operations analyzed. There was a drop in the insertion throughput with the increase in the number of organizations in the network, making the hypothesis feasible. Whereas in the search, the throughput remained constant, matching the frequency of arrival of requests. Through this analysis, the increase in the number of organizations worsens the blockchain network performance for data insertions, with a drop in throughput and an increase in average latency.

7. Conclusion

This study assessed the use of blockchain for heterogeneous data storage and distribution. The paper outlines two experiments focusing on aspects of blockchain networks: block creation, endorsement between nodes, and network structure. The first experiment outlines the validation and block creation process within the Hyperledger Fabric blockchain network. Our results showed that validating a transaction before its insertion into the network does not impact the analyzed metrics concerning the block creation process. This experiment highlights the significance of the block creation process in Fabric's network in scenarios involving heterogeneous data. The results show a reduction in latency by up to 6441.98% and an increase in throughput by up to 83.33% when comparing various block sizes and block creation times. Moreover, the experiment achieved superior performance, displaying an 83.01% decrease in latency and a 73.68% increase in throughput for insertion requests of 10,000 compared to the baseline of 1,000 requests in [Spengler and Souza 2021a], both with an arrival rate of 10 requests per second. This comparison underscores the impact of the block creation process with heterogeneous data.

The second experiment shows the influence of network size and network division into organizations on the Fabric's performance. Our outcomes reveal that the number of nodes within the blockchain network does affect its performance. As the number of nodes increases, both throughput and latency performance worsens. The results indicate that dividing the blockchain network into more organizations worsens performance. This division increased latency by up to 180.09% and decreased throughput by up to 24.58% in our evaluated scenarios. The runs for this second experiment were configured while considering the aspects outlined in all prior experiments. The data flow analysis in this experiment considered a network's peak service per node with 15,000 requests and an arrival rate of 50 requests/sec. with database usage. The block creation process was based on the best-case scenario involving a block size of 500 transactions and a creation time of 10s. The utilization of the database explored its impact on performance when network nodes increase. Considering all these factors, the block creation process emerges as a pivotal determinant of Fabric's blockchain network performance when dealing with heterogeneous data. Careful consideration of these factors is crucial, as the right choices can mitigate the impact on application performance, even when using the database and scaling with high write request rates and increased network nodes.

In future work, we will explore the validation process in networks with more nodes and organizations than those analyzed in this paper. This approach must provide a clearer understanding of the validation step's impact on blockchain application performance.

References

Al-Sumaidaee, G., Alkhudary, R., Zilic, Z., and Swidan, A. (2023). Performance analysis of a private blockchain network built on hyperledger fabric for healthcare. *Information Processing Management*, 60(2):103160.

Antonopoulos, A. M. (2017). *Mastering Bitcoin: Programming the Open Blockchain*. O'Reilly Media, Inc., 2nd edition.

Baliga, A., Solanki, N., Verekar, S., Pednekar, A., Kamat, P., and Chatterjee, S. (2018). Performance characterization of hyperledger fabric. In *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*, pages 65–74.

Chung, G., Desrosiers, L., Gupta, M., Sutton, A., Venkatadri, K., Wong, O., and Zugic, G. (2019). Performance tuning and scaling enterprise blockchain applications.

Dinh, T. T. A., Wang, J., Chen, G., Liu, R., Ooi, B. C., and Tan, K.-L. (2017). Blockbench: A framework for analyzing private blockchains. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pages 1085–1100, New York, NY, USA. ACM.

Fang, F., Ventre, C., Basios, M., Kanthan, L., Martinez-Rego, D., Wu, F., and Li, L. (2022). Cryptocurrency trading: a comprehensive survey.

Foundation, A. S. (2020). Apache couchdb. Acessado em 13/08/2023.

Honar Pajooh, H., Rashid, M. A., Alam, F., and Demidenko, S. (2022). Experimental performance analysis of a scalable distributed hyperledger fabric for a large-scale iot testbed. *Sensors*, 22(13).

Hyperledger (2019). hyperledger-fabricdocs documentation. Acessado em 13/08/2023.

Johnson, A. E., Pollard, T. J., Shen, L., Lehman, L. H., Feng, M., Ghassemi, M., Moody, B., Szolovits, P., Celi, L. A., and Mark, R. G. (2016). Mimic-iii, a freely accessible critical care database. *Sci Data 3*.

Project, C. (2020). Hyperledger caliper documentation. Acessado em 13/08/2020.

Roehrs, A., da Costa, C. A., da Rosa Righi, R., da Silva, V. F., Goldim, J. R., and Schmidt, D. C. (2019). Analyzing the performance of a blockchain-based personal health record implementation. *Journal of Biomedical Informatics*, 92:103140.

Shen, B., Guo, J., and Yang, Y. (2019). Medchain: Efficient healthcare data sharing via blockchain. *Applied Sciences*, 9(6).

Spengler, A. C. and Souza, P. S. (2021a). Avaliação de desempenho do hyperledger fabric com banco de dados para o armazenamento de grandes volumes de dados médicos. In *Anais do XX Workshop em Desempenho de Sistemas Computacionais e de Comunicação*, pages 61–72, Porto Alegre, RS, Brasil. SBC.

Spengler, A. C. F. and Souza, P. S. L. d. (2021b). The impact of using couchdb on hyperledger fabric performance for heterogeneous medical data storage. In *2021 XLVII Latin American Computing Conference (CLEI)*, pages 1–10.